KENYON & KENYON
Intellectual Property Law

RiverPark Towers
333 West San Carlos Street
Suite 600
San Jose, CA 95110
Direct Tel: (408) 975-7970
Facsimile: (408) 975-7501
sneal@kenyon.com

FACSIMILE TRANSMITTAL SHEET

TO: Examiner: Lilian VO	FROM: Stephen T. Neal
COMPANY: USPTO	DATE: January 30, 2006
FAX NUMBER: (571) 273-8300	TOTAL NO. OF PAGES INCLUDING COVER: 19
PHONE NUMBER:	SENDER'S REFERENCE NUMBER: Intel 2207/12020
RE: Serial No.: 10/047,809	YOUR REFERENCE NUMBER: Group Art Unit: 2195

☐ URGENT ■ FOR REVIEW ☐ PLEASE COMMENT ☐ PLEASE REPLY ☐ CONFIRMATION

☐ ORIGINAL WILL FOLLOW ■ ORIGINAL WILL NOT FOLLOW

Notes/Comments: **APPEAL BRIEF**

1. Fax Cover Sheet (1)
2. Fee Transmittal Form (and one copy) (2)
3. Appeal Brief (16)

Total: (17) pages

Certificate of Facsimile Transmittal

I hereby certify that the above referenced correspondence is being transmitted via facsimile under 37 C.F.R. §1.116 to Examiner: Lilian VO at facsimile number: (571) 273-8300 at the United States Patent and Trademark Office.

Dated: January 30, 2006

Signature: Barbara Vance
Barbara Vance

81895_1.DOC

PTO/SB/17 (10-04v2)
Approved for use through 07/31/2006. OMB 0651-0032
U.S. Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

FEE TRANSMITTAL for FY 2005

Effective 10/01/2004. Patent fees are subject to annual revision.

☐ Applicant claims small entity status. See 37 CFR 1.27

TOTAL AMOUNT OF PAYMENT ($) 500.00

Complete if Known	
Application Number	10/047,809
Filing Date	January 15, 2002
First Named Inventor	Ken SHOEMAKER et al.
Examiner Name	Lilian VO
Art Unit	2195
Attorney Docket No.	Intel 2207/12020

METHOD OF PAYMENT *(check all that apply)*

☐ Check ☐ Credit card ☐ Money Order ☐ Other ☐ None

☒ Deposit Account:

Deposit Account Number: 11-0600

Deposit Account Name: Kenyon & Kenyon LLP

The Director is authorized to: *(check all that apply)*
☒ Charge fee(s) indicated below ☒ Credit any overpayments
☒ Charge any additional fee(s) or any underpayment of fee(s)
☒ Charge fee(s) indicated below, except for the filing fee to the above-identified deposit account.

FEE CALCULATION

1. BASIC FILING FEE

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
1001	790	2001	395	Utility filing fee	
1002	350	2002	175	Design filing fee	
1003	550	2003	275	Plant filing fee	
1004	790	2004	395	Reissue filing fee	
1005	160	2005	80	Provisional filing fee	
				SUBTOTAL (1)	($) 0

2. EXTRA CLAIM FEES FOR UTILITY AND REISSUE

			Extra Claims		Fee from below		Fee Paid
Total Claims		-20 ** =		X	50.00	=	
Independent Claims		-3 ** =		X	200.00	=	
Multiple Dependent				X		=	

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description
1202	50	2202	25	Claims in excess of 20
1201	200	2201	100	Independent claims in excess of 3
1203	360	2203	180	Multiple dependent claim, if not paid
1204	200	2204	100	** Reissue independent claims over original patent
1205	50	2205	25	** Reissue claims in excess of 20 and over original patent

SUBTOTAL (2) ($) 0

**or number previously paid, if greater; For Reissues, see above

FEE CALCULATION (continued)

3. ADDITIONAL FEES

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
1051	130	2051	65	Surcharge - late filing fee or oath	
1052	50	2052	25	Surcharge - late provisional filing fee or cover sheet.	
1053	130	1053	130	Non-English specification	
1812	2,520	1812	2,520	For filing a request for *ex parte* reexamination	
1804	920*	1804	920*	Requesting publication of SIR prior to Examiner action	
1805	1,840*	1805	1,840*	Requesting publication of SIR after Examiner action	
1251	120	2251	60	Extension for reply within first month	
1252	450	2252	225	Extension for reply within second month	
1253	1,020	2253	510	Extension for reply within third month	
1254	1,590	2254	795	Extension for reply within fourth month	
1255	2,160	2255	1,080	Extension for reply within fifth month	
1401	500	2401	250	Notice of Appeal	
1402	500	2402	250	Filing a brief in support of an appeal	500.00
1403	1,000	2403	500	Request for oral hearing	
1451	1,510	1451	1,510	Petition to institute a public use proceeding	
1452	500	2452	250	Petition to revive – unavoidable	
1453	1,500	2453	750	Petition to revive – unintentional	
1501	1,400	2501	685	Utility issue fee (or reissue)	
1502	490	2502	245	Design issue fee	
1503	660	2503	330	Plant issue fee	
1460	130	1460	130	Petitions to the Commissioner	
1807	50	1807	50	Processing fee under 37 CFR 1.17 (q)	
1806	180	1806	180	Submission of Information Disclosure Stmt	
8021	40	8021	40	Recording each patent assignment per property (times number of properties)	
1809	790	2809	395	Filing a submission after final rejection (37 CFR § 1.129(a))	
1810	790	2810	395	For each additional invention to be examined (37 CFR § 1.129(b))	
1801	790	2801	395	Request for Continued Examination (RCE)	
1802	900	1802	900	Request for expedited examination of a design application	
				Other fee (specify) ____	

*Reduced by Basic Filing Fee Paid

SUBTOTAL (3) ($) 500.00

SUBMITTED BY

				Complete (if applicable)	
Name (Print/Type)	Stephen T. Neal	Registration No. (Attorney/Agent)	47,815	Telephone	(408) 975-7500
Signature	Stephen T. Neal			Date	January 30, 2006

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

This collection of information is required by 37 CFR 1.17 and 1.27. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing this form, call 1-800-PTO-9199 (1-800-786-9199) and select option 2.

RECEIVED
CENTRAL FAX CENTER
JAN 30 2006

Patent

Attorney Docket No.: Intel 2207/12020
U.S. Serial No.: 10/047,809
Assignee: Intel Corporation

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPELLANTS	:	Ken SHOEMAKER et al.
SERIAL NO.	:	10/047,809
FILED	:	January 15, 2002
FOR	:	APPARATUS AND METHOD FOR SCHEDULING THREADS IN MULTI-THREADING PROCESSORS
GROUP ART UNIT	:	2195
EXAMINER	:	Lilian VO

VIA FACSIMILE

M/S: APPEAL BRIEF - PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being facsimile transmitted to the United States Patent and Trademark Office via facsimile number (571) 273-8300 or deposited with the United States Postal Service as first class mail in an envelope addressed to: M/S APPEAL BRIEF - PATENTS, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on

Dated: January 30, 2006 Signature: Barbara Vance

Barbara Vance

ATTENTION: Board of Patent Appeals and Interferences

APPEAL BRIEF

Dear Sir:

This brief is in furtherance of the Notice of Appeal, filed in this case on November 29, 2005.

02/01/2006 EFLORES 00000096 110600 10047809
01 FC:1402 500.00 DA

Application No.: 10/047,809
Date: January 30, 2006
APPEAL BRIEF – PATENTS

1. REAL PARTY IN INTEREST

The real party in interest in this matter is Intel Corporation. (Recorded April 18, 2002, Reel/Frame 012839/0327).

2. RELATED APPEALS AND INTERFERENCES

There are no related appeals.

3. STATUS OF THE CLAIMS

Claims 1-20 are pending in the application. Claims 1, 9, and 15 stand rejected under 35 U.S.C. §102(b). Claims 2-8, 10-14, and 16-20 stand rejected under 35 U.S.C. §103(a). This appeal is an appeal from the rejection of claims 1-23.

4. STATUS OF AMENDMENTS

All claims amended in the previous response were admitted by the Examiner.

5. SUMMARY OF THE CLAIMED SUBJECT MATTER

Independent claim 1 recites a multi-threading processor. A first instruction fetch unit receives a first thread and a second instruction fetch unit to receive a second thread. (*See* p.7, lines 27-28 and elements 16 and 18 of Fig. 2). An execution unit executes the first thread and the second thread. (*See* p.8, lines 9-11 and element 26 of Fig. 2). A multi-thread scheduler coupled to the first instruction fetch unit, the second instruction fetch unit, and the execution unit, determines the width of said execution unit. (*See* p.8, lines 9-11 and element 24 of Fig. 2).

Independent claim 9 recites a computer implemented method for scheduling threads in a multithreading processor. A multi-threading processor is determined to be wide enough to execute a first thread and a second thread in parallel. (*See* p.9, lines 9-11 and element 32 of Fig. 3). The first thread and the second thread are executed in parallel if the multi-threading processor is wide enough to execute the first thread and the second thread in parallel. (*See* p.9, lines 12-14 and element 34 of Fig. 3).

Independent claim 15 recites a set of instructions residing in a storage medium to implement a method for scheduling threads in a multithreading processor. A multi-threading processor is determined to be wide enough to execute a first thread and a second thread in parallel. (*See* p.9, lines 9-11 and element 32 of Fig. 3). The first thread and the second thread are executed in parallel if the multi-threading processor is wide enough to execute the first thread and the second thread in parallel. (*See* p.9, lines 12-14 and element 34 of Fig. 3).

6. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

A. Claims 1, 9, and 15 were rejected under 35 U.S.C. §102(b) as being anticipated by "Simultaneous Multithreading: A Platform for Next-Generation Processor" by Eggers et al., (hereinafter "Eggers").

B. Claim 2 was rejected under 35 U.S.C. §103(a) as being unpatentable over Eggers.

C. Claims 3-8, 10-14, and 16-20 were rejected under 35 U.S.C. §103(a) as being unpatentable over Eggers in view of what the Examiner refers to as Applicants' (in this case Appellants') admitted prior art, (hereinafter "AAPA").

7. ARGUMENT

A. Claims 1, 9, and 15 were rejected under 35 U.S.C. §102(b) as being anticipated by Eggers.

Claims 1, 9, and 15 were rejected under 35 U.S.C. §102(b) as being anticipated by Eggers. Independent claim 1 recites a multi-threading processor. A first instruction fetch unit receives a first thread and a second instruction fetch unit to receive a second thread. An execution unit executes the first thread and the second thread. A multi-thread scheduler coupled to the first instruction fetch unit, the second instruction fetch unit, and the execution unit, determines the width of said execution unit.

As to claim 1, the Examiner has asserted that Eggers teaches the first instruction fetch unit and the second instruction fetch unit coupled to a multi-thread scheduler, referring to page 14, left column of Eggers. Appellants respectfully disagree because Eggers employs only one instruction fetch unit.

A SMT fetch unit of Eggers partitions instruction bandwidth among threads. The unit has eight program counters, one for each thread context. On each cycle, it selects two different threads and fetches eight instructions from each thread, and chooses a subset of these instructions for decoding (Eggers, page 14, line 3 from the bottom of the left column to line 3 of the right column). Eggers' thread selection hardware gives highest priority to the threads with the fewest instructions in the decoding, renaming and queue pipeline stages (Eggers, page 14, right column, lines 24-27). Thus, Eggers uses only one fetch unit to select threads. Eggers fails to teach the recited first and second instruction fetch unit. It is improper for the Examiner to read both the first instruction fetch unit and the second instruction fetch unit of claim 1 on the single fetch unit of Eggers.

The Examiner has also asserted that Eggers teaches the multi-thread scheduler which is to determine the width of an execute unit, as recited in claim 1. Appellants respectfully disagree. Eggers talks about simultaneous multithreading, but fails to teach the multi-thread scheduler which is to determine the width of an execution unit. Eggers uses dynamic scheduling hardware in current out-of-order superscalars for simultaneous multithreaded scheduling (Eggers, page 13, the last full paragraph of the right column). Eggers also states:

> In each cycle, the processor fetches eight instructions from the instruction cache. After instruction decoding, the register renaming logic maps the architectural registers to the hardware renaming registers to remove false dependencies. Instructions are then fed to either the integer or floating-point dispatch queues. When their operands become available, instructions are issued from these queues to their corresponding functional units. To support out-of-order execution, the processor tracks instruction and operand dependencies so that it can determine which instructions it can issue and which must wait for previously issued instructions to finish.

(Eggers, page 13, the paragraph bridging the columns).

Thus, in Eggers, the instructions are issued when their operands become available. Eggers does not mention width of the execution unit at all.

Accordingly, Appellants respectfully resubmit that claim 1 is patentable.

Independent claims 9 and 15 recite a computer implemented method and instructions for a computer implemented method. A multi-threading processor is determined to be wide enough to execute a first thread and a second thread in parallel. The first thread and the second thread are executed in parallel if the multi-threading processor is wide enough to execute the first thread and the second thread in parallel.

As to claims 9 and 15, the Examiner has asserted that the step of determining whether the processor is wide enough to execute the first and second threads in parallel is inherent in Eggers,

arguing that because Eggers' processor exploits both thread-level and instruction-level parallelism and schedules to execute multiple threads together, the processor width must have been taken into account to determine whether the processor is wide enough to schedule and execute those threads in parallel. Appellants respectfully disagree.

To establish inherency, the extrinsic evidence " 'must make clear that the missing descriptive matter is necessarily present in the thing described in the reference, and that it would be so recognized by persons of ordinary skill. Inherency, however, may not be established by probabilities or possibilities. The mere fact that a certain thing may result from a given set of circumstances is not sufficient.'" *In re Robertson*, 169 F.3d 743 (Fed. Cir. 1999).

Here, the purpose of Eggers is to reduce horizontal and vertical waste of processors. Eggers does not address the opposite problem: what if the processor finds more instructions than it can execute in a cycle. Even assuming that the opposite problem would be recognized by a skilled artisan, there are other solutions, for example, by limiting the instruction width of the issue hardware, or by increasing the width of the processor.

Thus, it is not necessary for Eggers to determine whether the multi-threading processor is wide enough to execute the first and second thread in parallel, and the Examiner's assertion that this feature is inherent in Eggers is not supported.

Accordingly, claims 9 and 15 are patentable over Eggers.

Accordingly reconsideration and withdrawal of the rejection of claims 1, 9, and 15 under 35 U.S.C. §102(b) is respectfully requested.

B. Claim 2 was rejected under 35 U.S.C. §103(a) as being unpatentable over Eggers.

Claim 2 stands rejected under 35 U.S.C. §103(a) as being unpatentable over Eggers. As stated above, claim 2 is patentable over Eggers, because several features of claim 2 are missing from Eggers.

C. Claims 3-8, 10-14, and 16-20 were rejected under 35 U.S.C. §103(a) as being unpatentable over Eggers in view of the AAPA

Claims 3-8, 10-14, and 16-20 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Eggers in view of the AAPA. As stated above, Eggers fails to teach or suggest the invention of claim 1, claim 9 and claim 15. The AAPA does not supply the deficiencies. Consequently, dependent claims 3-8, 10-14 and 16-20 are patentable.

The Examiner has argued that Eggers does not teach away from using an in-order processor because the processor of Eggers is able to schedule multiple threads when each have low instruction-level parallelism. Appellants respectfully disagree.

An in-order machine does not include hardware to determine instruction dependency, and instructions are executed in the same order that a compiler or program places them (Specification, page 3, the second full paragraph). However, Eggers uses dynamic scheduling hardware in current out-of-order superscalars for simultaneous multithread scheduling (Eggers, page 13, the first and second full paragraphs of the right column). Accordingly, Appellants respectfully resubmit that Eggers teaches away from using an in-order processor, and is not a proper §103 reference of claims 3-8, 11-14, and 17-20.

It is also impropriate for the Examiner to combine Eggers, which uses out-of-order superscalars, with AAPA, which talks about in-order processors. Appellants respectfully submit that there is no suggestion or motivation to combine the background of the application and Eggers beyond the impermissible use of hindsight. The Examiner references p. 13, 6th paragraph of Eggers as providing a motive to combine, which states:

> We derived our SMT model from a high-performance, out-of-order, superscalar architecture whose dynamic scheduling core is similar to that of the Mips R10000. In each cycle the processor fetches eight instruction from the instruction cache. After instruction decoding, the register-renaming logic maps the architectural registers to the hardware renaming registers to remove false dependencies. Instructions are then fed to either the integer or floating-point dispatch queues. When their operands become available, instructions are issued from these queues to their corresponding functional units. To support out-of-order execution, the processor tracks instruction and operand dependencies so that it can determine which instructions it can issue and which must wait for previously issued instructions to finish. After instructions complete execution, the processor retires them in order and frees hardware registers that are no longer needed.

(Eggers, p. 13, paragraph 6).

Eggers clearly provides no such motive to combine, as the portion cited by the Examiner clearly deals with out-of-order processors rather than the in-order processors of this application. Such motive would have to come from the AAPA, meaning that the motive combine came from the Appellants own insight.

Therefore, Appellants submit that a *prima facie* case of obviousness has not been made. The MPEP requires that the references must suggest making the combinations. MPEP §2141.01 (citing Hodosh v. Block Drug Co., Inc.); §706.02(j) (the initial burden is on the examiner to provide a convincing line of reasoning with explicit or implicit suggestions to combine references).

Merely stating that it would have been obvious for a person of ordinary skill in the art to combine references, without pointing to a specific hint or suggestion to combine, has been rejected by the Federal Circuit, as an invalid basis of rejection under 35 U.S.C. §103. *In re Lee*, 277 F.3d 1338, 1343 (Fed. Cir. 2002) (the court rejected a conclusory statement that it would have been obvious to combine the references without evidence of a teaching, motivation, or suggestion to select and combine the references, citing numerous cases); *In re Dembiczak*, 175 F.3d 994,999 (Fed. Cir. 1999) ("Our case law makes clear that the best defense against the subtle but powerful attraction of a hindsight-based obviousness analysis is rigorous application of the requirement for a showing of the teaching or motivation to combine prior art references.") It would be impossible for these references to be combined minus hindsight. Any motive to combine present in the background section of the application would be from the applicant or hindsight and not the result of any prior art.

Appellants therefore respectfully request that the Board of Patent Appeals and Interferences reverse the Examiner's decision rejecting claims 1-20 and direct the Examiner to pass the case to issue.

Application No.: 10/047,809
Date: January 30, 2006
APPEAL BRIEF – PATENTS

The Examiner is hereby authorized to charge any additional fees, which may be necessary for consideration of this paper to Kenyon & Kenyon Deposit Account No. **11-0600.**

Respectfully submitted,

KENYON & KENYON LLP

Date: January 30, 2006

By: _____
Stephen T. Neal
(Reg. No. 47,815)

KENYON & KENYON LLP
333 West San Carlos St., Suite 600
San Jose, CA 95110

Telephone: (408) 975-7500
Facsimile: (408) 975-7501

APPENDIX

(Brief of Appellant Nicholas G. SAMRA et al.
U.S. Patent Application Serial No. 09/752,573)

8. CLAIMS ON APPEAL

1. A multi-threading processor, comprising:

a first instruction fetch unit to receive a first thread and a second instruction fetch unit to receive a second thread;

an execution unit to execute said first thread and said second thread; and

a multi-thread scheduler coupled to said first instruction fetch unit, said second instruction fetch unit, and said execution unit, wherein said multi-thread scheduler is to determine the width of said execution unit.

2. A multi-threading processor as recited in claim 1, wherein the multi-thread scheduler unit determines whether the execution unit is to execute the first thread and the second thread in parallel depending on the width of the execution unit.

3. A multi-threading processor as recited in claim 2, wherein the multi-thread processor is an in-order processor.

4. A multi-threading processor as recited in claim 3, wherein the execution unit executes the first thread and the second thread in parallel.

5. A multi-threading processor as recited in claim 3, wherein the execution unit executes the first thread and the second thread in series.

6. A multi-threading processor as recited in claim 3, wherein the first thread and the second thread are compiled to have instruction level parallelism.

7. A multi-threading processor as recited in claim 6, further comprising:

a first instruction decode unit coupled between the first instruction fetch unit and the multi-thread scheduler; and

a second instruction decode unit coupled between the second instruction fetch unit and the multi-thread scheduler.

8. A multi-threading processor as recited in claim 4, wherein the execution unit executes only two threads in parallel.

9. A computer implemented method, comprising:

determining whether a multi-threading processor is wide enough to execute a first thread and a second thread in parallel; and

executing said first thread and said second thread in parallel if said multi-threading processor is wide enough to execute the first thread and the second thread in parallel.

10. The method as recited in claim 9, further comprising executing the first thread and the second thread in series if said multi-threading processor is not wide enough.

11. The method as recited in claim 10, wherein the multi-threading processor is an in-order processor.

12. The method as recited in claim 11, further comprising compiling the first thread and the second thread, wherein the first thread and the second thread have instruction level parallelism.

13. The method as recited in claim 12, wherein the multi-threading processor executes only two threads in parallel.

14. The method as recited in claim 13, further comprising:

fetching the first thread and the second thread; and

decoding the first thread and the second thread.

15. A set of instructions residing in a storage medium, said set of instructions to be executed by a multi-threading processor for searching data stored in a mass storage device comprising:

determining whether said multi-threading processor is wide enough to execute a first thread and a second thread in parallel; and

executing said first thread and said second thread in parallel if said multi-threading processor is wide enough to execute the first thread and the second thread in parallel.

16. A set of instructions as recited in claim 15, further comprising executing the first thread and the second thread in series if said multi-threading processor is not wide enough.

17. A set of instructions as recited in claim 16, wherein the multi-threading processor is an in-order processor.

18. A set of instructions as recited in claim 17, further comprising compiling the first thread and the second thread, wherein the first thread and the second thread have instruction level parallelism.

19. A set of instructions as recited in claim 18, wherein the multi-threading processor executes only two threads in parallel.

20. A set of instructions as recited in claim 19, further comprising:

fetching the first thread and the second thread; and

decoding the first thread and the second thread.

9. EVIDENCE APPENDIX

No further evidence has been submitted with this Appeal Brief.

10. RELATED PROCEEDINGS APPENDIX

Per Section 2 above, there are no related proceedings to the present Appeal.